LAPORAN PRAKTIKUM DATA STRUCTURE

HASH TABLES

Dosen Pengampu:

H. Fatchurrochman,M.Kom

Asisten Praktikum:

Fillah Anjany 230605110033

Oleh :

Muhammad Alif Mujaddid **240605110082**

Jurusan Teknik Informatika Fakultas Sains dan Teknologi

UIN Maulana Malik Ibrahim Malang

2025

A. KESIMPULAN

Kesimpulan yang diperoleh dari pembahasan praktikum kali ini adalah:

1. Tentang Hash Table

   Hash table merupakan salah satu struktur data yang dirancang untuk melakukan operasi penyimpanan, pencarian, dan penghapusan data secara sangat efisien. Struktur ini memanfaatkan fungsi hash untuk mengubah suatu data atau key menjadi nilai indeks tertentu, sehingga data dapat langsung ditempatkan atau diakses tanpa melalui proses pencarian berulang seperti pada array atau linked list biasa. Prinsip ini membuat hash table memiliki kompleksitas waktu rata-rata *O(1)* untuk operasi insert, search, dan delete, asalkan distribusi data merata. Penggunaan hash table sangat relevan dalam bidang komputasi yang memerlukan efisiensi tinggi, seperti database indexing, caching, pencocokan kata pada dictionary, dan implementasi struktur set atau map pada bahasa pemrograman.

2. Tentang Perbedaan Open Addressing dan Separate Chain

   Collision merupakan masalah yang tidak dapat dihindari dalam hash table, terutama ketika jumlah elemen yang disimpan mendekati ukuran tabel. Untuk mengatasinya, digunakan dua pendekatan utama: open addressing dan separate chaining.

   Pada open addressing, seluruh data disimpan di dalam tabel utama. Jika terjadi collision, algoritma akan melakukan pencarian posisi alternatif (slot kosong) menggunakan teknik probing tertentu. Metode ini tidak menggunakan memori tambahan, tetapi performanya dapat menurun saat tabel semakin penuh.

   Pada separate chaining, setiap indeks tabel dapat menampung lebih dari satu elemen dengan menggunakan struktur data tambahan, umumnya linked list. Dengan cara ini, setiap elemen yang memiliki nilai hash sama akan dikelompokkan dalam rantai yang sama. Metode ini lebih fleksibel karena tabel tidak akan "penuh", namun membutuhkan alokasi memori tambahan untuk setiap rantai penyimpanan.

3. Tentang perbedaan linier probing, quadratic probing, dan double hashing sebagai collision resolution

   Ketiga teknik tersebut merupakan variasi metode probing dalam open addressing untuk menangani collision.

   Linear probing melakukan pencarian slot kosong secara berurutan (misalnya +1, +2, +3, ...). Metode ini sederhana dan mudah diimplementasikan, namun mudah menyebabkan *primary clustering*, yaitu penumpukan data di area tertentu sehingga memperlambat proses pencarian.

   Quadratic probing memperbaiki kelemahan linear probing dengan memberikan jarak pencarian yang berbentuk kuadrat (misalnya $+1^2$, $+2^2$, $+3^2$, ...). Pola non-linier ini membantu mengurangi penumpukan data, namun tetap memiliki potensi *secondary clustering* jika fungsi hash kurang baik.

Double hashing menggunakan dua fungsi hash yang berbeda. Ketika terjadi collision, nilai hash kedua digunakan untuk menentukan langkah pergeseran, sehingga pergerakan pencarian tidak bergantung pada pola tetap. Teknik ini memberikan distribusi paling merata dan memiliki tingkat keberhasilan tinggi dalam mengurangi clustering, meskipun implementasinya lebih kompleks.

Secara keseluruhan, praktikum ini memberikan pemahaman bahwa pemilihan metode penanganan collision sangat bergantung pada kebutuhan aplikasi, kapasitas memori, dan tingkat efisiensi yang diharapkan. Hash table tetap menjadi salah satu struktur data yang paling efektif untuk operasi berbasis pencarian cepat, namun keberhasilannya sangat ditentukan oleh desain fungsi hash dan strategi collision handling yang digunakan.